CyclicJudge: Mitigating Judge Bias Efficiently in LLM-based Evaluation

Anonymous ACL submission

Abstract

LLM-as-judge evaluation has become standard practice for open-ended model assessment; however, judges exhibit systematic biases that cannot be eliminated by increasing the number of scenarios or generations. These biases are often similar in magnitude to the model differences that benchmarks are designed to detect, resulting in unreliable rankings when single-judge evaluations are used. This work introduces a variance decomposition that partitions benchmark score variance into scenario, generation, judge, and residual components. Based on this analysis, **CyclicJudge**, a round-robin assignment of judges to scenarios, is demonstrated to be the optimal strategy for a fixed judge-call budget. It eliminates bias precisely while requiring each judge only once per cycle, matching the cost of single-judge evaluation. Empirical results on MT-Bench and MindEval validate the effectiveness of CyclicJudge as predicted, across both general-purpose and domain-specific evaluation settings.

1 Introduction

LLM-as-judge evaluation has become the de facto standard for open-ended model assessment (Zheng et al., 2023). A growing body of work indicates that LLM judges are systematically biased: some models are consistently lenient, others strict, with additional position, verbosity, and self-preference effects (Zheng et al., 2023; Shi et al., 2024; Wang et al., 2024). A standard Analysis of Variance (ANOVA; Fisher, 1925) on our data confirms that judge main effect is highly significant for all evaluated models (§4.2).

Crucially, systematic bias does not average away. Adding more test scenarios or regenerating responses reduces random noise but leaves judge bias intact. The only way to eliminate judge bias is to use multiple judges whose biases cancel. Yet running every judge on every item multiplies evaluation cost by the panel size and sacrifices generation diversity at a fixed budget.

We show that **CyclicJudge**, a round-robin assignment of judges to items, resolves this tension. To establish this formally, we develop a variance decomposition rooted in generalizability theory (Brennan, 2001; Shavelson and Webb, 1991) that separates benchmark score variance into scenario, generation, judge, and residual components. Our contributions are:

1. A mixed-effects model that decomposes benchmark variance into random noise and systematic judge bias, showing they require fundamentally different remedies (§3.1–3.2).
2. A proof that round-robin cycling achieves lower variance than alternative strategies: it eliminates bias while maximizing generation diversity at any budget (§3.3).
3. Empirical results on MT-Bench and MindEval validating the effectiveness of CyclicJudge as predicted, across both general-purpose and domain-specific evaluation settings (§4).

2 Related Work

LLM judge biases and multi-judge approaches. LLM judges exhibit systematic position bias (Wang et al., 2024; Shi et al., 2024), length bias (Dubois et al., 2024), self-preference (Wataoka et al., 2024), and cognitive biases (Koo et al., 2024); we refer readers to Gu et al. (2024) for a comprehensive survey. Multi-judge panels mitigate these biases by averaging over diverse judge tendencies (Verga et al., 2024), while psychometric reliability analyses further demonstrate that single-judge evaluations yield insufficiently stable scores (Schroeder and Wood-Doughty, 2024).

Benchmark reliability and generation variance. Madaan et al. (2025) empirically quantifies variance across hundreds of models and benchmarks,

and Alvarado Gonzalez et al. (2025) shows via ANOVA that unrepeated evaluations frequently produce rank inversions across benchmark subcategories. Zhang et al. (2025) proposes a hierarchical model decomposing variance into within-prompt and between-prompt components, and Mizrahi et al. (2024) shows that prompt paraphrasing alone can reverse model rankings. Our work extends this line by further separating systematic judge bias from random noise.

Budget-constrained evaluation. Saha et al. (2026) frames judge selection as a multi-armed bandit problem, Luettgau et al. (2025) applies hierarchical Bayesian partial pooling in low-data regimes, and Miller (2024) derives standard errors and power formulas for benchmark scores; Dorner et al. (2025) further establishes fundamental scaling limits on the accuracy gains from additional judge queries.

3 Methodology

3.1 Model Specification

Consider evaluating model θ on n scenarios from a dataset $\mathcal{D}$ of N total scenarios, with m independent generations per scenario, each scored by K judges drawn from a pool of K_{tot}. The same K judges score every scenario–generation cell (a crossed design). We model each score as:

$$X_{ij\ell} = \mu_\theta + \alpha_i + \beta_{ij} + \gamma_\ell + \varepsilon_{ij\ell} \qquad (1)$$

where $i \in \{1, \ldots, n\}$ indexes scenarios, $j \in \{1, \ldots, m\}$ indexes generations within scenario i, and ℓ indexes the K selected judges. Each term has a distinct source:

- μ_θ: true ability of model θ (the estimand).
- $\alpha_i \overset{\text{iid}}{\sim} (0, \sigma_\alpha^2)$: scenario effect—how much individual scenario difficulty varies for model θ. σ_α^2 absorbs the scenario–model interaction; a dataset whose scenarios all test the same aspects of model θ equally has $\sigma_\alpha^2 = 0$.
- $\beta_{ij} \overset{\text{iid}}{\sim} (0, \sigma_\beta^2)$: generation effect—variability from stochastic decoding. σ_β^2 is model-specific; a deterministic model has $\sigma_\beta^2 = 0$.
- γ_ℓ: judge bias—a fixed constant for each judge, with centering constraint $\sum_{\ell=1}^{K_{\text{tot}}} \gamma_\ell = 0$ and population dispersion $\sigma_\gamma^2 = K_{\text{tot}}^{-1} \sum_\ell \gamma_\ell^2$. It is fixed rather than random because judge tendencies are persistent properties documented in prior work (Wang et al., 2024; Koo et al., 2024).
- $\varepsilon_{ij\ell} \overset{\text{iid}}{\sim} (0, \sigma_\varepsilon^2)$: residual—captures judge-level noise and all interaction terms; we discuss these modeling choices in Appendix I.

The random effects $(\alpha, \beta, \varepsilon)$ are mutually independent with finite second moments; no distributional assumptions (e.g. Gaussianity) are needed. The n scenarios and K judges are each drawn by simple random sampling without replacement, independently of each other and of all random effects.

3.2 Variance Decomposition

The benchmark score is the grand mean $\bar{X} = \frac{1}{nmK} \sum_{i,j,\ell} X_{ij\ell}$, which decomposes as $\bar{X} = \mu_\theta + \bar{\alpha} + \bar{\beta} + \bar{\gamma} + \bar{\varepsilon}$, where each bar denotes a hierarchical sample mean.

Proposition 1 (Variance decomposition)**.**

$$\mathrm{Var}(\bar{X}) = \underbrace{\frac{\sigma_\alpha^2}{n} + \frac{\sigma_\beta^2}{nm} + \frac{\sigma_\varepsilon^2}{nmK}}_{\textit{random noise}} + \underbrace{\frac{\sigma_\gamma^2}{K} \cdot \frac{K_{tot} - K}{K_{tot} - 1}}_{V_\gamma\textit{: judge bias}} \qquad (2)$$

Each noise term shrinks with more data at or above its hierarchy level. The bias term V_γ responds only to K and vanishes exactly at $K = K_{\text{tot}}$. The proof uses the law of total variance, conditioning on the judge selection, combined with a finite population correction for sampling K from K_{tot} centered bias values (Appendix C).

3.3 Allocation Strategies

Given a per-scenario budget of B judge calls, the design question is how to allocate judges across generations. The scenario term σ_α^2/n is fixed once scenarios are chosen; we compare three strategies for the remaining variance.

Strategy A: All judges per generation. Use all K_{tot} judges for each of $m = B/K_{\text{tot}}$ generations. Judge bias cancels exactly ($\bar{\gamma} = 0$):

$$V_A(B) = \frac{K_{\text{tot}}\, \sigma_\beta^2 + \sigma_\varepsilon^2}{nB} \qquad (3)$$

Strategy B: Random single judge per generation. Use $m = B$ generations, each scored by one randomly sampled judge. The random bias acts as additional noise:

$$V_B(B) = \frac{\sigma_\beta^2 + \sigma_\gamma^2 + \sigma_\varepsilon^2}{nB} \qquad (4)$$

Strategy C: CyclicJudge (round-robin). Use $m = B$ generations (B divisible by K_{tot}), assigning judge $j \bmod K_{\text{tot}}$ to generation j. Bias cancels exactly while generation count matches Strategy B:

$$V_C(B) = \frac{\sigma_\beta^2 + \sigma_\varepsilon^2}{nB} \tag{5}$$

With one generation per scenario ($m = 1$), the same variance and bias cancellation are achieved by cycling judges across scenarios instead: assign judge $i \bmod K_{\text{tot}}$ to scenario i (n divisible by K_{tot}).

Advantage of CyclicJudge. Direct comparison (Appendix E) shows $V_C \leq \min(V_A, V_B)$ for all B, with gaps of $O(1/B)$, so CyclicJudge's advantage is largest at small budgets of judge calls, where most benchmarks operate. The ranking of A vs. B depends on whether $(K_{\text{tot}}-1)\sigma_\beta^2 \gtrless \sigma_\gamma^2$: all judges beat random only when judge bias is large relative to generation variance.

4 Experiments

4.1 Domain and Data

We validate on two benchmarks that span different evaluation regimes.

MT-Bench (Zheng et al., 2023) is a general-purpose multi-turn conversational benchmark with 80 two-turn questions spanning 8 categories, scored on a 1–10 scale per turn (averaged).

MindEval (Pombal et al., 2025) is a domain-specific benchmark for multi-turn mental health support with 50 scenarios and composite scores on a 1–5 scale covering empathy, safety, and clinical appropriateness. Each scenario simulates a full 10-turn therapeutic session with a synthetic patient.

For both benchmarks, we evaluate five models—Qwen 2.5 7B Instruct (Yang et al., 2024), Llama 3.3 70B Instruct (Grattafiori et al., 2024), GPT-5.2, Gemini 3 Flash, and Claude Sonnet 4.6—which also serve as judges in a symmetric design. MT-Bench uses m=10 generations per scenario (4,000 scores per model); MindEval uses m=5 (1,250 scores per model). Variance components are estimated via crossed ANOVA (Appendix D).

4.2 Judge Bias Estimates

We test whether judge identity has a significant effect using a two-way ANOVA, treating each scenario–generation pair as a subject crossed with K judges. On both benchmarks, the judge main effect is highly significant for every evaluated model ($p < 0.001$ in all cases; see Appendix H).

Table 1: Mean score ± 95% CI at the default operating point (m=1, K=1). **Bold**: highest score per judge column. Underline: second highest.

(a) MT-Bench (1–10 scale, n=80)

	Judge				
Model	Qwen	Llama	GPT	Gemini	Claude
Gemini	8.02±.16	8.98±.08	8.20±.12	**9.88±.06**	**8.68±.10**
GPT	7.91±.16	8.81±.11	**8.67±.12**	9.71±.11	8.49±.13
Claude	7.96±.17	**9.06±.06**	7.91±.13	9.62±.12	8.52±.13
Llama	7.87±.18	8.71±.11	6.90±.20	8.58±.25	7.26±.22
Qwen	**8.02±.15**	8.39±.16	6.09±.20	7.16±.27	5.99±.23

(b) MindEval (1–5 scale, n=50)

	Judge				
Model	Qwen	Llama	GPT	Gemini	Claude
Claude	3.86±.08	**4.67±.07**	**4.05±.06**	**4.72±.06**	**4.84±.04**
GPT	**4.04±.09**	4.51±.07	3.76±.09	3.95±.16	3.39±.15
Gemini	3.90±.08	4.49±.06	3.63±.05	3.91±.12	3.31±.08
Llama	3.70±.07	4.29±.05	3.16±.06	3.00±.10	2.85±.06
Qwen	3.26±.12	3.38±.21	2.32±.08	1.69±.08	1.74±.05

Table 1 demonstrates that **single-judge evaluation produces unreliable rankings**. On MT-Bench at the default operating point (m=1, K=1), the Qwen judge ranks Qwen first despite being last by others, and the top-ranked model changes with every judge—a consequence of the self-preference biases documented by Wataoka et al. (2024). The per-observation variance estimates (Table 2) confirm this quantitatively: judge variance $\hat{\sigma}_\gamma^2$ ranges from 0.34 to 0.95 across models. Since competitive models on current leaderboards are routinely separated by < 0.5 points (Chiang et al., 2024), single-judge evaluation can easily reverse rankings. On MindEval, rankings are notably more stable: four of five judges produce an identical ordering. This stability arises because domain-specific rubrics leave less room for stylistic self-preference, so judge biases act as near-uniform shifts rather than model-dependent interactions (see Appendix A for per-judge bias patterns).

4.3 Variance Components

Table 2 reports the per-observation variance component estimates for both benchmarks. On MT-Bench, scenario variance $\hat{\sigma}_\alpha^2$ is the largest per-observation component; maximizing the number of distinct scenarios always reduces total variance (Appendix F). Generation variance $\hat{\sigma}_\beta^2$ drops with capability, consistent with the observation that **stronger models**

Table 2: Per-observation variance component estimates.

(a) MT-Bench (n=80)

Component	Qwen	Llama	GPT	Gemini	Claude
$\hat{\sigma}^2_\alpha$ (scenario)	1.530	0.882	0.634	0.408	0.393
$\hat{\sigma}^2_\beta$ (generation)	0.266	0.238	0.076	0.000	0.000
$\hat{\sigma}^2_\gamma$ (judge)	0.947	0.503	0.339	0.435	0.427
$\hat{\sigma}^2_\varepsilon$ (residual)	1.486	1.130	0.564	0.661	0.830

(b) MindEval (n=50)

Component	Qwen	Llama	GPT	Gemini	Claude
$\hat{\sigma}^2_\alpha$ (scenario)	0.034	0.014	0.021	0.015	0.005
$\hat{\sigma}^2_\beta$ (generation)	0.100	0.019	0.047	0.016	0.002
$\hat{\sigma}^2_\gamma$ (judge)	0.523	0.280	0.132	0.150	0.155
$\hat{\sigma}^2_\varepsilon$ (residual)	0.159	0.069	0.207	0.103	0.061

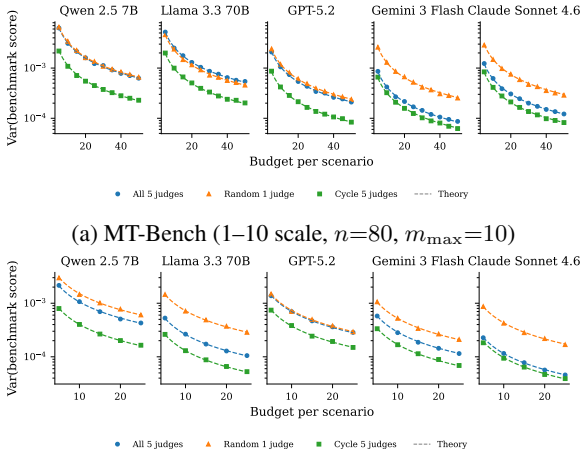
produce more deterministic outputs. As frontier models increasingly rely on RL post-training, generation variance may shrink further (Matsutani et al., 2025). Judge bias $\hat{\sigma}^2_\gamma$ remains substantial for all models, accounting for >94% of benchmark variance at the default operating point (see Appendix B).

On MindEval, both scenario and generation variance are markedly lower. Scenario variance $\hat{\sigma}^2_\alpha$ drops by one to two orders of magnitude because all MindEval scenarios evaluate the same rubrics across different user profiles, so difficulty varies far less than on MT-Bench. Generation variance is similarly suppressed: each MindEval generation is a full 10-turn therapeutic session, so per-response decoding randomness averages out over many turns. Judge bias remains the dominant component throughout, exceeding both scenario and generation variance by an order of magnitude.

4.4 Strategy Comparison

Figure 1 validates the three allocation strategies via 5,000 subsampling repetitions at each budget level on both benchmarks. Markers show empirical variance; dashed lines show exact predictions from empirical pool variances (Appendix G). CyclicJudge achieves lower variance everywhere on both benchmarks, and predictions match empirical results precisely. At B=5, switching from random to cycling cuts variance by ∼27–40% across models on MT-Bench.

The optimal strategy among the two alternatives depends on the sign of $(K_{\text{tot}}-1)\sigma^2_\beta-\sigma^2_\gamma$: all-judges wins when $\sigma^2_\gamma/\sigma^2_\beta > K_{\text{tot}}-1 = 4$. On MT-Bench, this ratio is below the threshold for Qwen (3.6) and Llama (2.1), so random judging is preferable; for Gemini and Claude ($\hat{\sigma}^2_\beta \approx 0$), all-judges converges



(a) MT-Bench (1–10 scale, n=80, $m_{\max}$=10)

(b) MindEval (1–5 scale, n=50, $m_{\max}$=5)

Figure 1: Benchmark score variance $\text{Var}(\bar{X})$ vs. per-scenario budget for three strategies: all judges (A), random single judge (B), and CyclicJudge (C).

to CyclicJudge. On MindEval, the ratio exceeds the threshold for nearly every model, so all-judges consistently outperforms random. The driver is not higher judge bias, but rather sharply lower generation variance $\hat{\sigma}^2_\beta$. Meanwhile, judge disagreement over subjective clinical criteria keeps $\hat{\sigma}^2_\gamma$ substantial relative to $\hat{\sigma}^2_\beta$. **CyclicJudge outperforms both alternatives regardless of this ratio**.

5 Conclusion

We develop a variance decomposition for LLM-as-judge evaluation that separates systematic judge bias from random noise and use it to derive CyclicJudge, a round-robin judge assignment strategy. Our analysis and experiments on both a general-purpose benchmark (MT-Bench) and a domain-specific clinical benchmark (MindEval) yield three main findings: (1) Systematic judge bias dominates benchmark score variance at default operating points. (2) Random noise components decrease with model capability and multi-turn evaluation, but judge bias does not. (3) CyclicJudge achieves the lowest variance among all allocation strategies at every budget level, regardless of the model's variance profile, by eliminating bias exactly while maximizing generation diversity. The method requires no model-specific tuning and maintains the same per-item cost as single-judge evaluation, providing practitioners with a cost-neutral drop-in replacement for more reliable LLM evaluation.

Limitations

Linear model approximation. Our variance decomposition assumes a linear random effects model, which treats scores as unbounded and continuous. LLM-as-judge scores are in fact bounded ordinal variables, so the linear model can produce boundary artifacts and heteroscedastic residuals near the scale endpoints. A generalized linear mixed model (GLMM), such as a cumulative-link mixed model, would properly account for the ordinal, bounded nature of the responses. However, GLMMs do not yield the closed-form variance decomposition that our allocation analysis relies on. We therefore retain the linear approximation, noting that it remains reasonable when the scale range is large relative to the observed score dispersion, as in our benchmarks.

Judge pool size. Our judge pool of $K_{\text{tot}} = 5$ limits exploration of larger panels. While the theoretical results generalize to arbitrary K, empirical validation with larger and more diverse judge pools would strengthen the practical recommendations.

Scenario exchangeability. The "maximize scenarios" recommendation assumes scenarios are interchangeable draws from the dataset; in practice, some scenarios may be more informative than others, and item-selection methods (Maia Polo et al., 2024) could alter the tradeoff.

Balanced design assumption. The allocation analysis assumes a balanced design in which every scenario receives the same number of judge calls. Extensions to adaptive or sequential designs remain future work.

Uniform cost assumption. Our budget analysis treats all judge calls as equal cost; in practice, judges vary widely in price and latency. A cost-weighted allocation that accounts for per-model pricing and inference speed would be needed for real deployment decisions.

References

Miguel Angel Alvarado Gonzalez, Michelle Bruno Hernandez, Miguel Angel Peñaloza Perez, Bruno Lopez Orozco, Jesus Tadeo Cruz Soto, and Sandra Malagon. 2025. Do repetitions matter? strengthening reliability in LLM evaluations. *Preprint*, arXiv:2509.24086.

Robert L. Brennan. 2001. *Generalizability Theory*. Springer, New York.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating LLMs by human preference. In *Proceedings of the 41st International Conference on Machine Learning*.

Florian E. Dorner, Vivian Y. Nastl, and Moritz Hardt. 2025. Limits to scalable evaluation at the frontier: LLM as judge won't beat twice the data. In *International Conference on Learning Representations*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2024. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. *Preprint*, arXiv:2404.04475.

Ronald A. Fisher. 1925. *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 537 others. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Yinghan Li, Jie Liu, Yiwen Li, Yicheng Shen, Jian Jiang, Shengyao Guo, and Pengjun Xie. 2024. A survey on LLM-as-a-judge. *Preprint*, arXiv:2411.15594.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2024. Benchmarking cognitive biases in large language models as evaluators. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Lennart Luettgau, Alexander Coppock, Yves Dubois, Christopher Summerfield, and Cozmin Ududec. 2025. HiBayES: A hierarchical Bayesian modeling framework for AI evaluation statistics. *Preprint*, arXiv:2505.05602.

Lovish Madaan, Samarth Singh, Rylan Schaeffer, Andrew Poulton, Sanmi Koyejo, Pontus Stenetorp, Sharan Narang, and Dieuwke Hupkes. 2025. Quantifying variance in evaluation benchmarks. In *International Conference on Learning Representations*.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. 2024. tinyBenchmarks: Evaluating LLMs with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*.

Kohsei Matsutani, Shota Takashiro, Gouki Minegishi, Takeshi Kojima, Yusuke Iwasawa, and Yutaka Matsuo. 2025. RL squeezes, SFT expands: A comparative study of reasoning LLMs. *Preprint*, arXiv:2509.21128.

Evan Miller. 2024. Adding error bars to evals: A statistical approach to language model evaluations. *Preprint*, arXiv:2411.00640.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? A call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12.

José Pombal, Maya D'Eon, Nuno M. Guerreiro, Pedro Henrique Martins, António Farinhas, and Ricardo Rei. 2025. MindEval: Benchmarking language models on multi-turn mental health support. *Preprint*, arXiv:2511.18491.

Aadirupa Saha, Aniket Wagde, and Branislav Kveton. 2026. LLM-as-judge on a budget. *Preprint*, arXiv:2602.15481.

Kayla Schroeder and Zach Wood-Doughty. 2024. Can you trust LLM judgments? reliability of LLM-as-a-judge. *Preprint*, arXiv:2412.12509.

Shayle R. Searle, George Casella, and Charles E. McCulloch. 1992. *Variance Components*. Wiley, New York.

Richard J. Shavelson and Noreen M. Webb. 1991. *Generalizability Theory: A Primer*. Sage Publications, Newbury Park, CA.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. 2024. Judging the judges: A systematic study of position bias in pairwise comparison with LLM-as-a-judge. *Preprint*, arXiv:2406.07791.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Leontiadis, Gautier Izacard, and Fabio Petroni. 2024. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models. *Preprint*, arXiv:2404.18796.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. 2024. Self-preference bias in LLM-as-a-judge. *Preprint*, arXiv:2410.21819.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 24 others. 2024. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Yifan Zhang, Dingwen Cai, and Xingjian Chen. 2025. Beyond the singular: Revealing the value of multiple generations in benchmark evaluation. *Preprint*, arXiv:2502.08943.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36.

A Judge Bias Estimates

Figure 2 shows the estimated per-judge bias for each judge–model pair on both benchmarks. A pronounced self-preference is visible along the diagonal: on MindEval, Claude Sonnet 4.6's self-bias of $+0.412$ is the *only* positive entry in its otherwise uniformly negative row, and Qwen 2.5 7B's self-bias is the largest in its row on both benchmarks.

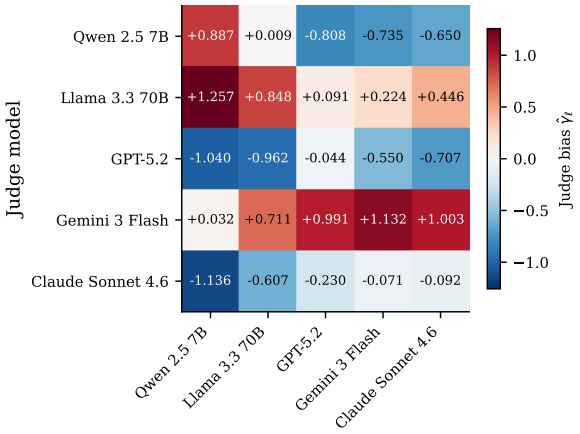
Judge temperament is not stable across benchmarks: Gemini 3 Flash is the most lenient judge on MT-Bench (all biases positive), yet becomes one of the harshest on MindEval, suggesting that judge personality is benchmark-dependent rather than an intrinsic model property. On MT-Bench, strictness correlates with model capability: the two strongest models by consensus (GPT-5.2 and Claude) are the most uniformly negative judges, while the two weakest (Qwen and Llama) are the most lenient. Finally, the harshest judge–model interactions concentrate on capability-mismatched pairs: GPT-5.2 and Claude penalize Qwen by over one full scale point on MT-Bench, while their biases toward each other are an order of magnitude smaller.

B Variance Component Estimates

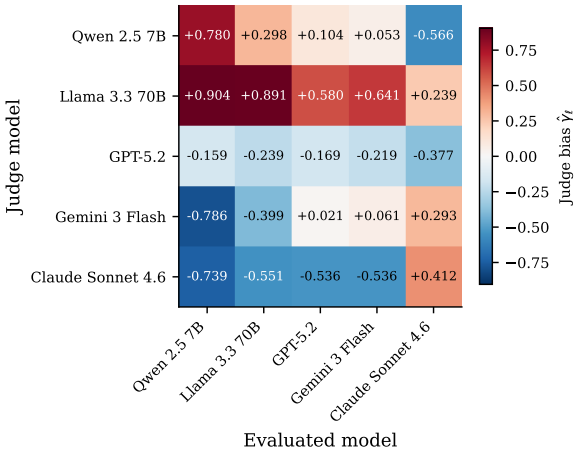
Table 3 reports the full variance component estimates. The generation variance entries $\hat{\sigma}^2_\beta = 0.000$ (Gemini and Claude on MT-Bench) arise because the method-of-moments estimator $\hat{\sigma}^2_\beta = (\mathrm{MS}_G - \mathrm{MS}_W)/K$ (Appendix D) can return a negative value when the between-generation mean square is smaller than the residual mean square. Because a variance is non-negative by definition, negative estimates are set to zero—the standard truncation convention for ANOVA-based variance component estimation (Searle et al., 1992). These zeros, therefore, indicate that generation-to-generation variability is negligible relative to residual noise for these models, not that it is exactly absent.

C Proof of Proposition 1

We prove the variance decomposition in three steps: (1) grand mean decomposition, (2) law of total variance, and (3) finite population correction for judge bias.



(a) MT-Bench



(b) MindEval

Figure 2: Estimated judge biases $\hat{\gamma}_\ell = \bar{X}_{..\ell} - \bar{X}_{...}$ for each judge–model pair.

Step 1: Grand mean decomposition. Starting from $X_{ij\ell} = \mu_\theta + \alpha_i + \beta_{ij} + \gamma_\ell + \varepsilon_{ij\ell}$, we average hierarchically.

Average over K judges within generation (i, j). The terms μ_θ, α_i, and β_{ij} are constant across judges:

$$\begin{aligned}\bar{X}_{ij\cdot} &= \frac{1}{K}\sum_{\ell \in S_J} X_{ij\ell} \\ &= \mu_\theta + \alpha_i + \beta_{ij} + \underbrace{\frac{1}{K}\sum_{\ell \in S_J}\gamma_\ell}_{\bar{\gamma}} + \underbrace{\frac{1}{K}\sum_{\ell \in S_J}\varepsilon_{ij\ell}}_{\bar{\varepsilon}_{ij}}\end{aligned} \tag{6}$$

The judge bias mean $\bar{\gamma}$ is the same for every (i, j) cell because the same S_J evaluates all cells (crossed design).

Table 3: Variance component estimates. *Per-observation variances* and their *contribution to* $\mathrm{Var}(\bar{X})$ at the default operating point (m=1, K=1).

(a) MT-Bench (n=80)

Component	**Qwen**	**Llama**	**GPT**	**Gemini**	**Claude**
$\hat{\mu}_\theta$ (mean score)	7.13	7.87	8.72	8.75	8.61
Per-observation variance					
$\hat{\sigma}^2_\alpha$ (scenario)	1.530	0.882	0.634	0.408	0.393
$\hat{\sigma}^2_\beta$ (generation)	0.266	0.238	0.076	0.000	0.000
$\hat{\sigma}^2_\gamma$ (judge)	0.947	0.503	0.339	0.435	0.427
$\hat{\sigma}^2_\varepsilon$ (residual)	1.486	1.130	0.564	0.661	0.830
Contribution to $\mathrm{Var}(\bar{X})$ *(*$\times 10^{-3}$*)*					
$\hat{\sigma}^2_\alpha/n$	19.13	11.03	7.92	5.11	4.92
$\hat{\sigma}^2_\beta/nm$	3.33	2.97	0.94	0.00	0.00
$\hat{\sigma}^2_\gamma \cdot \mathrm{FPC}/K$	946.50	502.50	339.00	435.40	426.60
$\hat{\sigma}^2_\varepsilon/nmK$	18.57	14.12	7.05	8.27	10.38

(b) MindEval (n=50)

Component	**Qwen**	**Llama**	**GPT**	**Gemini**	**Claude**
$\hat{\mu}_\theta$ (mean score)	2.478	3.400	3.931	3.846	4.431
Per-observation variance					
$\hat{\sigma}^2_\alpha$ (scenario)	0.034	0.014	0.021	0.015	0.005
$\hat{\sigma}^2_\beta$ (generation)	0.100	0.019	0.047	0.016	0.002
$\hat{\sigma}^2_\gamma$ (judge)	0.523	0.280	0.132	0.150	0.155
$\hat{\sigma}^2_\varepsilon$ (residual)	0.159	0.069	0.207	0.103	0.061
Contribution to $\mathrm{Var}(\bar{X})$ *(*$\times 10^{-3}$*)*					
$\hat{\sigma}^2_\alpha/n$	0.67	0.28	0.43	0.29	0.09
$\hat{\sigma}^2_\beta/nm$	2.01	0.38	0.93	0.31	0.04
$\hat{\sigma}^2_\gamma \cdot \mathrm{FPC}/K$	522.50	280.20	132.00	150.20	154.80
$\hat{\sigma}^2_\varepsilon/nmK$	3.18	1.37	4.14	2.05	1.23

Average over m generations within scenario i.

$$\bar{X}_{i\cdot\cdot} = \mu_\theta + \alpha_i + \underbrace{\frac{1}{m}\sum_{j=1}^{m}\beta_{ij}}_{\bar{\beta}_i} + \bar{\gamma} + \underbrace{\frac{1}{mK}\sum_{j,\ell}\varepsilon_{ij\ell}}_{\bar{\varepsilon}_{i\cdot}} \tag{7}$$

Average over n scenarios.

$$\bar{X} = \mu_\theta + \bar{\gamma} + \underbrace{\frac{1}{n}\sum_{i=1}^{n}\alpha_i}_{\bar{\alpha}} + \underbrace{\frac{1}{nm}\sum_{i,j}\beta_{ij}}_{\bar{\beta}} + \underbrace{\frac{1}{nmK}\sum_{i,j,\ell}\varepsilon_{ij\ell}}_{\bar{\varepsilon}} \tag{8}$$

Step 2: Law of total variance. Let $\mathcal{S}$ denote the judge selection. By the law of total variance:

$$\mathrm{Var}(\bar{X}) = \mathrm{E}\big[\mathrm{Var}(\bar{X} \mid \mathcal{S})\big] + \mathrm{Var}\big(\mathrm{E}[\bar{X} \mid \mathcal{S}]\big) \tag{9}$$

Conditioning on $\mathcal{S}$:

Term (I): Conditional variance. Given $\mathcal{S}$, the bias $\bar{\gamma}$ is a constant. The remaining randomness

comes from $\bar{\alpha}$, $\bar{\beta}$, $\bar{\varepsilon}$, which are means of mutually independent zero-mean random variables:

$$\mathrm{Var}(\bar{X} \mid \mathcal{S}) = \frac{\sigma_\alpha^2}{n} + \frac{\sigma_\beta^2}{nm} + \frac{\sigma_\varepsilon^2}{nmK} \tag{10}$$

This does not depend on $\mathcal{S}$, so $\mathrm{E}[\mathrm{Var}(\bar{X} \mid \mathcal{S})]$ equals the same expression.

Term (II): Variance of conditional mean. Since $\mathrm{E}[\alpha_i] = \mathrm{E}[\beta_{ij}] = \mathrm{E}[\varepsilon_{ij\ell}] = 0$:

$$\mathrm{E}[\bar{X} \mid \mathcal{S}] = \mu_\theta + \bar{\gamma} \implies \mathrm{Var}\big(\mathrm{E}[\bar{X} \mid \mathcal{S}]\big) = \mathrm{Var}(\bar{\gamma}) \tag{11}$$

Step 3: Finite population correction. $\bar{\gamma}$ is a sample mean of K values drawn without replacement from the centered population $\{\gamma_1, \ldots, \gamma_{K_{\text{tot}}}\}$.

Lemma 2 (Finite population correction)**.** *Let $a_1, \ldots, a_P$ satisfy $\sum_i a_i = 0$ with $\sigma_a^2 = P^{-1}\sum_i a_i^2$. If S is a simple random sample of size k drawn without replacement:*

$$\mathrm{Var}\left(\frac{1}{k}\sum_{i \in S} a_i\right) = \frac{\sigma_a^2}{k} \cdot \frac{P-k}{P-1} \tag{12}$$

Proof. Define indicator variables $Z_i \in \{0, 1\}$ for $i = 1, \ldots, P$, where $Z_i = 1$ if and only if element i is selected in the sample S. The sample sum is $T = \sum_{i=1}^{P} Z_i a_i$, and the sample mean is $\bar{a}_S = T/k$.

Under simple random sampling without replacement (SRSWOR) of size k from P elements, each element is selected with probability $\Pr(Z_i = 1) = k/P$, so:

$$\mathrm{Var}(Z_i) = \frac{k}{P}\left(1 - \frac{k}{P}\right) = \frac{k(P-k)}{P^2} \tag{13}$$

For $i \neq j$, the joint selection probability is $\Pr(Z_i{=}1,\ Z_j{=}1) = \frac{k(k-1)}{P(P-1)}$, giving:

$$\begin{aligned}\mathrm{Cov}(Z_i, Z_j) &= \mathrm{E}[Z_i Z_j] - \mathrm{E}[Z_i]\,\mathrm{E}[Z_j] \\ &= \frac{k(k-1)}{P(P-1)} - \frac{k^2}{P^2} = -\frac{k(P-k)}{P^2(P-1)}\end{aligned} \tag{14}$$

Since $T = \sum_i Z_i a_i$ and the a_i are constants, expanding the variance gives:

$$\begin{aligned}&\mathrm{Var}(T) \\ &= \sum_{i=1}^{P} a_i^2\ \mathrm{Var}(Z_i) + 2\sum_{i<j} a_i a_j\ \mathrm{Cov}(Z_i, Z_j) \\ &= \frac{k(P-k)}{P^2}\sum_i a_i^2 - \frac{k(P-k)}{P^2(P-1)} \cdot 2\sum_{i<j} a_i a_j\end{aligned} \tag{15}$$

We now simplify each factor. By definition, $\sum_i a_i^2 = P\sigma_a^2$. For the cross-term, the centering constraint $\sum_i a_i = 0$ implies:

$$\begin{aligned}0 = \Big(\sum_i a_i\Big)^2 &= \sum_i a_i^2 + 2\sum_{i<j} a_i a_j \\ \implies 2\sum_{i<j} a_i a_j &= -P\sigma_a^2\end{aligned} \tag{16}$$

Substituting both identities into (15) (the two negatives in the second term yield a positive contribution):

$$\begin{aligned}\mathrm{Var}(T) &= \frac{k(P-k)}{P^2} \cdot P\sigma_a^2 + \frac{k(P-k)}{P^2(P-1)} \cdot P\sigma_a^2 \\ &= \frac{k(P-k)\,\sigma_a^2}{P-1}\end{aligned} \tag{17}$$

Finally, dividing by k^2:

$$\mathrm{Var}(\bar{a}_S) = \frac{\mathrm{Var}(T)}{k^2} = \frac{\sigma_a^2}{k} \cdot \frac{P-k}{P-1} \tag{18}$$

□

The FPC factor equals 1 when $k = 1$ and 0 when $k = P$ (biases cancel exactly). Applying to judge biases ($a_\ell = \gamma_\ell$, $P = K_{\text{tot}}$, $k = K$):

$$\mathrm{Var}(\bar{\gamma}) = \frac{\sigma_\gamma^2}{K} \cdot \frac{K_{\text{tot}} - K}{K_{\text{tot}} - 1} \tag{19}$$

Combining Terms (I) and (II) completes the proof.

D Estimation of Variance Components

Because judges are crossed (not nested), we use a crossed ANOVA.

Residual noise. The residual mean square estimates σ_ε^2:

$$\begin{aligned}\hat{\sigma}_\varepsilon^2 &= \mathrm{MS}_W \\ &= \frac{\sum_{i,j,\ell}(X_{ij\ell} - \bar{X}_{ij\cdot} - \bar{X}_{\cdot\cdot\ell} + \bar{X}_{\cdot\cdot\cdot})^2}{(nm-1)(K-1)}\end{aligned} \tag{20}$$

Generation variance. The between-generation mean square:

$$\mathrm{MS}_G = \frac{K}{n(m-1)}\sum_{i,j}(\bar{X}_{ij\cdot} - \bar{X}_{i\cdot\cdot})^2 \tag{21}$$

with $\mathrm{E}[\mathrm{MS}_G] = \sigma_\varepsilon^2 + K\sigma_\beta^2$. Judge bias drops out because the same judges evaluate every generation within a scenario.

Scenario variance. The between-scenario mean square:

$$\mathrm{MS}_S = \frac{mK}{n-1} \sum_i (\bar{X}_{i\cdot\cdot} - \bar{X}_{\cdots})^2 \tag{22}$$

with $\mathrm{E}[\mathrm{MS}_S] = \sigma_\varepsilon^2 + K\sigma_\beta^2 + mK\,\sigma_\alpha^2$.

Method-of-moments estimators.

$$\hat{\sigma}_\beta^2 = (\mathrm{MS}_G - \mathrm{MS}_W)/K \tag{23}$$

$$\hat{\sigma}_\alpha^2 = (\mathrm{MS}_S - \mathrm{MS}_G)/(mK) \tag{24}$$

Judge bias. We estimate $\hat{\gamma}_\ell = \bar{X}_{\cdot\cdot\ell} - \bar{X}_{\cdots}$ and compute:

$$\hat{\sigma}_\gamma^2 = \frac{1}{K_{\text{tot}}} \sum_\ell \hat{\gamma}_\ell^2 - \frac{\hat{\sigma}_\varepsilon^2}{nm} \cdot \frac{K_{\text{tot}} - 1}{K_{\text{tot}}} \tag{25}$$

E Allocation Strategy Derivations

We derive the benchmark score variance for each strategy. The n scenarios are fixed, so σ_α^2/n contributes a constant offset and is excluded.

Strategy A: All K_{tot} judges per generation. Budget $B = mK_{\text{tot}}$, so $m = B/K_{\text{tot}}$. Per-scenario mean:

$$\bar{X}_i = \mu_\theta + \alpha_i + \frac{1}{m}\sum_j \beta_{ij} + \underbrace{\frac{1}{K_{\text{tot}}}\sum_\ell \gamma_\ell}_{=0} + \frac{1}{mK_{\text{tot}}}\sum_{j,\ell} \varepsilon_{ij\ell} \tag{26}$$

Judge bias cancels exactly. The benchmark variance from generation and residual noise:

$$V_A = \frac{1}{n}\left(\frac{\sigma_\beta^2}{m} + \frac{\sigma_\varepsilon^2}{mK_{\text{tot}}}\right) = \frac{K_{\text{tot}}\sigma_\beta^2 + \sigma_\varepsilon^2}{nB} \tag{27}$$

Strategy B: Random single judge per generation. Budget $B = m$, each generation scored by $\ell_j \sim \mathrm{Uniform}(\{1, \ldots, K_{\text{tot}}\})$. Per-scenario mean:

$$\bar{X}_i = \mu_\theta + \alpha_i + \frac{1}{m}\sum_j \big(\beta_{ij} + \gamma_{\ell_j} + \varepsilon_{ij\ell_j}\big) \tag{28}$$

Each term in the sum is independent with variance $\sigma_\beta^2 + \sigma_\gamma^2 + \sigma_\varepsilon^2$:

$$V_B = \frac{\sigma_\beta^2 + \sigma_\gamma^2 + \sigma_\varepsilon^2}{nB} \tag{29}$$

Strategy C: Round-robin cycling. Budget $B = m$ (multiple of K_{tot}), generation j scored by judge $j \bmod K_{\text{tot}}$. Each judge appears m/K_{tot} times, so bias cancels:

$$\frac{1}{m}\sum_{j=1}^{m} \gamma_{j \bmod K_{\text{tot}}} = \frac{1}{K_{\text{tot}}}\sum_{\ell=1}^{K_{\text{tot}}} \gamma_\ell = 0 \tag{30}$$

The residuals for different j are independent with variance σ_ε^2:

$$V_C = \frac{\sigma_\beta^2 + \sigma_\varepsilon^2}{nB} \tag{31}$$

Pairwise comparisons. Direct subtraction yields:

$$V_A - V_C = \frac{(K_{\text{tot}} - 1)\sigma_\beta^2}{nB} \geq 0 \tag{32}$$

$$V_B - V_C = \frac{\sigma_\gamma^2}{nB} \geq 0 \tag{33}$$

$$V_A - V_B = \frac{(K_{\text{tot}} - 1)\sigma_\beta^2 - \sigma_\gamma^2}{nB} \tag{34}$$

F Scenario Coverage

The strategies above fix n and optimize the per-scenario budget. A complementary question is how to split a total generation budget $B_{\text{gen}} = nm$ between scenarios and generations per scenario. Under cycling, substituting $n = B_{\text{gen}}/m$ into Eq. 5 gives:

$$V_C(B_{\text{gen}}, m) = \frac{m\,\sigma_\alpha^2 + \sigma_\beta^2 + \sigma_\varepsilon^2}{B_{\text{gen}}} \tag{35}$$

This is linear in m with positive slope $\sigma_\alpha^2/B_{\text{gen}}$, so variance is minimized at $m = 1$, maximizing the number of distinct scenarios up to the dataset size N. The same monotonicity holds for Strategies A and B. Once the scenario pool is exhausted ($n = N$), excess budget should go to additional generations.

G Exact Bootstrap Predictions

The allocation formulas in §3.3 assume infinite-population generation draws. When validating via bootstrap subsampling from a finite pool of $m_{\max}$ observed generations, we compute exact predictions using the empirical pool variances. For all three strategies, the subsampling observations within each scenario are i.i.d. draws from the relevant pool, so $V(B) = C/(nB)$ where C is the pool variance.

Strategy A. Each draw is a generation-level mean $\bar{X}_{ig\cdot}$, with $m = B/K_{\text{tot}}$ draws per scenario. The pool variance is:

$$C_A = K_{\text{tot}} \cdot \frac{1}{n} \sum_i \frac{1}{m_{\max}} \sum_g (\bar{X}_{ig\cdot} - \bar{X}_{i\cdot\cdot})^2 \quad (36)$$

Strategy B. Each draw is a single cell $X_{ig\ell}$ with generation and judge chosen uniformly at random, with $m = B$ draws per scenario:

$$C_B = \frac{1}{n} \sum_i \frac{1}{m_{\max} K_{\text{tot}}} \sum_{g,\ell} (X_{ig\ell} - \bar{X}_{i\cdot\cdot})^2 \quad (37)$$

Strategy C. Each judge ℓ gets B/K_{tot} independent draws from its column; the per-scenario mean averages these K_{tot} independent sub-means. The relevant pool variance is the within-column generation variance:

$$C_C = \frac{1}{n K_{\text{tot}}} \sum_{i,\ell} \frac{1}{m_{\max}} \sum_g (X_{ig\ell} - \bar{X}_{i\cdot\ell})^2 \quad (38)$$

These empirical constants converge to the population formulas (Eqs. 3–5) as $m_{\max} \to \infty$, but for finite $m_{\max}$ they account for the bias in plug-in variance estimates (the $(m_{\max}-1)/m_{\max}$ correction) and the fact that the crossed ANOVA residual uses global judge centering while C_C uses within-scenario-judge centering.

H ANOVA F-Test for Judge Effects

We test for a systematic judge effect using a two-way crossed ANOVA that treats each scenario–generation pair as a subject crossed with K judges. The judge sum of squares $\text{SS}_J = N \sum_{\ell=1}^{K} (\bar{X}_{\cdot\cdot\ell} - \bar{X}_{\cdot\cdot\cdot})^2$ (where $N = nm$) is compared to the residual $\text{SS}_R = \text{SS}_T - \text{SS}_J - \text{SS}_S$, yielding $F = \text{MS}_J/\text{MS}_R$ with $K-1$ and $(N-1)(K-1)$ degrees of freedom. The p-value is the survival function of the F-distribution, computed via `scipy.stats.f.sf`. Table 4 reports results for both benchmarks; in all cases, the null hypothesis of no judge effect is rejected at $p < 0.001$.

I Omitted Interactions

A full factorial design with three facets (scenario, generation, judge) has $2^3 = 8$ terms: a grand mean, three main effects, three two-way interactions, and one three-way interaction. Our model (Eq. 1) retains the grand mean and three main effects, absorbing all interaction terms into the residual $\varepsilon_{ij\ell}$. We enumerate each omitted interaction below.

Table 4: ANOVA F-statistic for the judge main effect. All $p < 0.001$.

Model	MT-Bench $F(4, 3196)$	MindEval $F(4, 996)$
Qwen 2.5 7B	638.2	1029.5
Llama 3.3 70B	445.8	1280.1
GPT-5.2	601.7	200.1
Gemini 3 Flash	659.2	458.0
Claude Sonnet 4.6	514.9	788.9

Scenario–generation. Because generations are nested within scenarios rather than crossed with them, there is no separable scenario–generation interaction: any such effect is already confounded with the generation main effect β_{ij}, which captures all within-scenario generation variability by construction.

Scenario–judge. This interaction captures the extent to which a judge's bias varies across scenarios (e.g., a judge being lenient on reasoning tasks but strict on creative ones). It is in principle estimable, but explicitly modeling it would add $n(K_{\text{tot}}-1)$ parameters and complicate the closed-form variance decomposition without changing the allocation analysis: the interaction enters only through the per-cell residual variance, which our empirical predictions (§4) already capture without parametric assumptions on its structure.

Generation–judge. This interaction reflects judge-specific reactions to particular generation artifacts (e.g., one judge penalizing verbose outputs more than another). Since generations are independent stochastic draws, this term is exchangeable across j and acts as additional i.i.d. noise on each cell, making it naturally absorbed into σ_ε^2.